

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVA

FRANJO GALEKOVIĆ

SVG GRAFIKE NA WEB-U

ZAVRŠNI RAD

ČAKOVEC, 2017.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVA

FRANJO GALEKOVIĆ

SVG GRAFIKE NA WEB-U
SVG GRAPHICS ON WEB

ZAVRŠNI RAD

Mentor:

dr. sc. Sanja Brekalo

ČAKOVEC, 2017.

Sažetak:

U ovom radu opisan je SVG (engl. Scalable Vector Graphics) format i način rada web aplikacija koje ga koriste, animacija SVG-a, dostupne biblioteke, 3D SVG grafike, prednosti i nedostaci. Prikazani su primjeri izrade jednostavnih oblika, dodavanja teksta, stila i transformacije, primjenjivanje gradijenata, isječaka, maskiranja i filtera, te kreiranje jednostavnog uzorka. Prikazana je i animacija, interaktivnost i skriptiranje sa SVG elementima. U nekoliko primjera odvijaju se dvije vrste animacija: koristeći SMIL (engl. Synchronized Multimedia Integration Language) i koristeći JavaScript. JavaScript je bolje podržan i više se koristi na web-u. SVG je XML (engl. eXtensible Markup Language) što u prijevodu znači: jezik za opis vektorske grafike, animacije i aplikacija. Važno je sučelje DOM (engl. Document Object Model) kojeg izrađuje pretraživač kad pročita SVG datoteku, jer se preko njega iscrtavaju i manipuliraju SVG datoteke. Nakon navedenih primjera, prikazana je funkcionalna web aplikacija: interaktivna SVG karta Hrvatske. Karta se sastoji od županija Hrvatske koje se mogu označiti i kao takve pohraniti u zadanom formatu. Izrađena aplikacija SVG karta Hrvatske može služiti kao smjernica posjećenih županija ili u neke edukativne svrhe. Aplikaciju je lako koristiti te je ista podržana u svim novijim verzijama pretraživača. Također, predstaviti će se dostupne biblioteke SVG-a, a neke od njih su: Raphaël - JavaScript Library, SVGWeb, dojox.gfx, SVGKit, amMap, PlotKit, JSDrawing, VectorGraphics i PERGOLA. Adobe Illustrator, Inkscape i SVG-edit su jedni od najpoznatijih SVG editora koji služe za izradu i uređivanje SVG grafike. Najvažnija karakteristika SVG dokumenata je njihova skalabilnost - moguće ih je proizvoljno smanjivati ili povećavati bez gubitka kvalitete ili izobličenja. Prema standardima koji važe u XML-u, SVG dokument se sastoji od hijerarhijski posloženih zapisa koji sadrže informacije o geometrijskim elementima ili tekstu od kojih se sastoji grafika. Glavne tehnologije za izradu web animacije su HTML5 Canvas, SVG i Flash. Danas se sve više koriste HTML5 Canvas i SVG tehnologije.

Ključne riječi: SVG, XML, animacija, JavaScript, DOM

Sadržaj

1. Uvod	5
2. Vektorski XML zapis	6
2.1. Razvoj tekstualnih vektorskih formata i nastanak SVG-a	6
2.2. SVG dokument	6
3. Animiranje SVG-a	9
4. 3D SVG grafike	10
4.1. SVG kao oznaka HTML5	10
4.2. Jednostavni objekti	10
4.3. Transformacije u 3D prostoru uz pomoć CSS3 standarda	10
5. SVG editori	11
5.1. Inkscape	11
5.2. Adobe Illustrator	12
5.3. SVG-Edit	12
6. Dostupne biblioteke	13
6.1. Raphaël - JavaScript Library	13
6.2. SVGWeb	13
6.3. Dojox.gfx	13
6.4. SVGKit	13
6.5. amMap	14
6.6. PlotKit	14
6.7. JSDrawing	14
6.8. VectorGraphics	14
6.9. PERGOLA	14
7. Web aplikacije	15
8. Priprema za rad sa SVG grafikom	16
9. Izrada osnovnih vektorskih oblika	17
9.1. Linije	17
9.2. Pravokutnik	18
9.3. Krug i elipsa	19
9.4. Izlomljene linije i poligoni	20
9.5. Putanja	21
9.6. Uzorci	22

10. Dodavanje teksta, stila i transformacije.....	23
10.1 Dodavanje i pozicioniranje teksta	23
10.2 Gradijenti.....	24
10.3. Isječci	25
10.4. Maskiranje	26
10.5. Transformacije.....	27
11. Animacije	29
11.1. Interakcija s animacijom.....	31
11.2. Skriptiranje	32
12. Filteri	36
13. Interaktivna SVG karta Hrvatske	38
14. Zaključak.....	43
Literatura	44

1. Uvod

U početku su web stranice koristile rastersku grafiku tijekom pojavljivanja prvih grafičkih web preglednika, dok se vektorska grafika koristila za: tablice, granice između odjeljaka, prekriženih ili podvučenih tekstova i slično. Budući da se vektorska grafika malo koristila, ubrzo je bila zamjenjivana rasterskom grafikom, čak i na mjestima gdje to nije bilo najbolje rješenje. U posljednjem je desetljeću dogovoren novi vektorski tekstualno baziran format koji je namijenjen web upotrebi - Scalable Vector Graphics (SVG). Uz ovaj format stvorena je mogućnost jednostavnog i kvalitetnog kreiranja vektorskih grafika koje su neovisne o platformi i dostupne su svakome za korištenje. SVG je jezik otvorenog standarda za izradu vektorske grafike na web-u s visokom rezolucijom te mogućnostima kao što su: animacija, filtri i gradijenti. Kako bi se izradile SVG datoteke, moguće je koristiti obične tekst editore poput Notepad i Notepad++. SVG pruža kvalitetan grafički dizajn s mogućnostima oblikovanja rasporeda grafičkih elemenata, interaktivnosti, izbora fontova i boja. SVG se bazira na XML-u te su moguće i dinamičke interaktivnosti. Zbog kompatibilnosti sa skriptnim jezicima, JavaScript-om je omogućeno pristupanje i manipuliranje svim SVG elementima.

Glavne tehnologije za izradu web animacija uz SVG su još i: HTML5 Canvas i Flash. HTML `<canvas>` element je oznaka koja se pojavila u verziji HTML5 i koristi se za iscrtavanje grafičkih elemenata, fotografskih kompozicija i izradi animacija pomoću JavaScript-a. Upotrebom JavaScript-a pristupa se čitavom setu funkcija za crtanje linija, krugova, kvadrata, putanja, teksta i dodavanja slika. Adobe Flash je alat koji se koristi za izradu vektorske grafike, animacije, aplikacija, prezentacija i brojnih drugih sadržaja za multimedijalnu interakciju s korisnikom. Ovaj program je prvenstveno vektorski program s mogućnošću implementacije rasterske grafike. U počecima izrade web animacije, nije bilo alternative Flashu, dok se danas sve više koriste HTML5 Canvas i SVG tehnologije.

2. Vektorski XML zapis

2.1. Razvoj tekstualnih vektorskih formata i nastanak SVG-a

World Wide Web Consortium (W3C) je na samom kraju 20. stoljeća osnovao SVG radnu grupu kako bi pronašao alternativu PostScript formatu koji koristi skalabilne fontove i objekte. Sredinom 1998. god. Adobe, IBM, Netscape i Sun Microsystems su predložili PGML (engl. Precision Graphics Language), dok su Hewlett Pacard, Macromedia, Microsoft i Visio predložili VML (engl. Vector Markup Language). Odlučilo se da će novi format biti optimiziran za web te da će se XML koristiti kao zapis. SVG standard se razvijao pod okriljem velikih korporacija te je zamišljen kao otvoreni standard, a sam razvoj nadzirao je W3C konzorcij, najvažnije tijelo za standardizaciju i razvoj raznih mrežnih tehnologija. XML je uzet kao baza SVG-a. SVG osim mogućnosti prikaza grafike za web stranice, ima ogroman broj i drugih mogućih primjena.[1]

2.2. SVG dokument

SVG dokument mora poštovati neke XML standarde, poput: strukturiranosti, valjanosti te zadana pravila koja su propisana SVG standardom. Korištenje XML-a ima svoje prednosti i nedostatke.

Prednosti:

- u običnom tekstualnom editoru osoba ga može lako pročitati
- podržava Unicode, na svim poznatim jezicima i na taj način omogućuje prikaz teksta
- pogodna hijerarhijska struktura za opisivanje sadržaja
- sadržaj se opisuje oznakama
- stroga sintaksna pravila omogućuju jednostavnije analiziranje podataka
- prihvaćen je kao međunarodni standard
- lako se može čitati na svakoj platformi

Nedostaci:

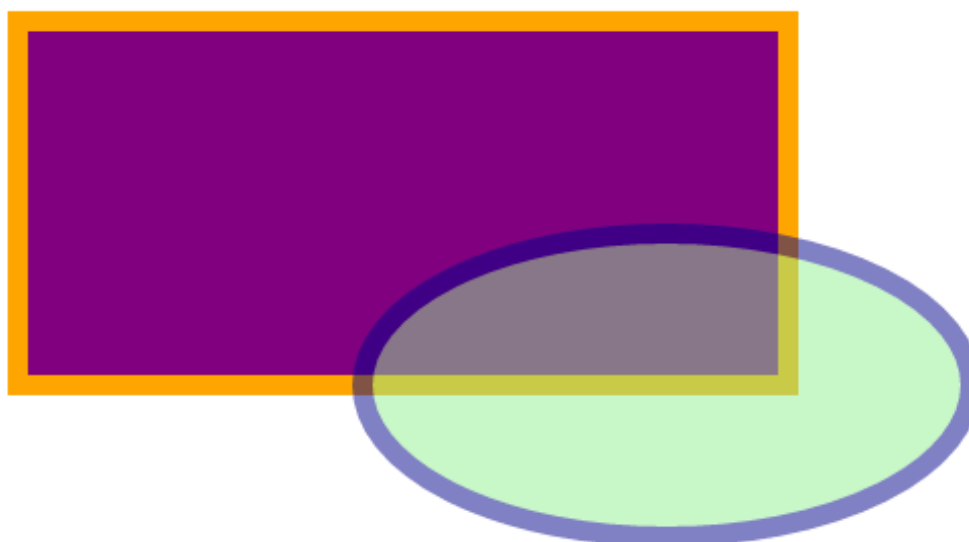
- sintaksa je opširna i redundantna (istovremeno postoje formati koji su manje redundantni, a ljudi ih lako mogu pročitati)
- hijerarhijska struktura pogodna je samo za neke strukture podataka

- ponekad nije očita razlika između atributa i sadržaja elemenata

SVG podržava skup osnovnih oblika: krug, pravokutnik, elipsa, linija i poligon. Osnovni oblici mogu se dobiti korištenjem krivulja. Krivulje se smatraju najvažnijim oblikom u SVG formatu jer rade na principu trenutne točke. Pozicija (0,0) je početna točka krivulje i do svake sljedeće zadane točke se povlači i iscrtava dio krivulje i on ovisi o zadanim parametrima krivulje i njenom putu. Iscrtavanje sadržaja se odvija u prostoru koji opisuje SVG platno. Crtanje je moguće u svim dimenzijama jer je platno beskonačno, no iscrtavanje se ipak odvija u nekom pravokutnom prostoru zvanom prozor, a isti može biti implicitno određen veličinom oblika ili u vanjskom dokumentu koji sadrži cijeli SVG dokument ili može biti zadan u prvom <svg> elementu. Prilikom inicijalizacije prozora korisnički koordinatni sustav i koordinatni sustav prozora su jednaki. Ishodište oba sustava se nalazi u gornjem lijevom uglu prozora, a pozitivno povećavanje vrijednosti koordinate je u desnom smjeru i prema dolje. Korištenjem novog <svg> elementa može se inicijalizirati novi koordinatni sustav koji u sebi može sadržavati informacije o novom položaju. Svi elementi unutar tog <svg> elementa neće računati poziciju prema osnovnom koordinatnom sustavu prozora, već prema <svg> elementu. Nakon prve transformacije elementa mijenja se korisnički koordinatni sustav. Moguće je primijeniti više transformacija svakom elementu. (Transformacija je preslikavanje korisničkih koordinata elementa u koordinate SVG prozora.) Podržane transformacije su: rotacija, translacija, smik - uzdužna transformacija i skaliranje - promjena mjerila.[1]

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">
  <rect x="30" y="30" width="380" height="180"
        fill="purple" stroke="orange" stroke-width="10" />
  <ellipse cx="350" cy="210" rx="150" ry="75" opacity="0.5"
          fill="lightgreen" stroke="darkblue" stroke-width="10" />
</svg>
```

Primjer 1: Jednostavni SVG dokument



Slika 2. Prikaz primjera 1. u web pregledniku s podrškom za SVG

Izvor: autor

3. Animiranje SVG-a

Mogućnost animacije imaju svi statički elementi u SVG-u. Postoje dva načina koji su podržani specifikacijom. Jedan od načina je pomoću SMIL-a (engl. Synchronized Multimedia Integration Language). SMIL je jezik baziran na XML-u, kojim se upravlja multimedijalnim sadržajima kao što su: audio, tekst, video i grafika na web-u. Neke od mogućnosti SMIL-a su: specifikacija rasporeda elemenata i sinkronizacija, rezolucija ekrana i spektar boje, prilagodba protoka informacija. Upotreba SMIL-a je rijetka i zapravo ne postoji neka veća ili vrlo raširena aplikacija ili tehnologija koja ga koristi. Budući da je slaba podrška SMIL-a u web preglednicima, inženjeri se više posvećuju animaciji pomoću ECMAScripta. ECMAScript je interpretirani jezik koji se najčešće zove i JavaScript. JavaScript je dominantni jezik za skriptiranje klijentske strane na web-u. Podržan je od strane mnogih poznatih web preglednika, a ujedno ga podržavaju i web preglednici na mobilnim uređajima. Standardiziranost i raširenost JavaScript-a, doprinijeli su stvaranju interaktivnosti, animacije te ostalih manipulacija SVG-a. JavaScript može dinamički mijenjati sadržaj. Web preglednici su dugo vremena prezentirali HTML te je zbog toga stvoren DOM (Document Object Model), i na taj način se sučelje standardiziralo. DOM je neovisan o platformi i jeziku za prikaz XML-a, HTML-a i sličnih formata. Web preglednici nisu dužni prikazivati HTML pomoću DOM-a, ali većina najpoznatijih preglednika se većim dijelom oslanja na DOM.[1]

4. 3D SVG grafike

4.1. SVG kao oznaka HTML5

SVG koristi vektorsku grafiku, što znači da koristeći objekte crta grafičke oblike, stoga je moguće svakom objektu izmijeniti pojedini atribut. Objekti zauzimaju vrlo malo memorije jer računalno u memoriju zapisuje samo vrijednosti atributa objekata, te su i datoteke koje sadržavaju elemente vektorske grafike, u principu male. Slika ne gubi na kvaliteti promjenom veličine ili zumiranjem objekata, što znači da je SVG neovisan o rezoluciji ekrana. SVG omogućuje prikaz samo 2D elemenata i transformacije nad njima, no mogu se raditi i 3D transformacije uz pomoć CSS3 standarda.[2]

4.2. Jednostavni objekti

Unaprijed definirani elementi koje sadrži SVG predstavljaju određene oblike. To su elementi: `<circle>`, `<rect>`, `<line>`, `<ellipse>`, `<path>`, `<polyline>`, `<polygon>` i `<text>`. Takvi elementi se mogu prikazati i u 3D obliku. Svaki element je objekt s vlastitim atributima, a vrijednosti elementa se zadaju prilikom crtanja.[2]

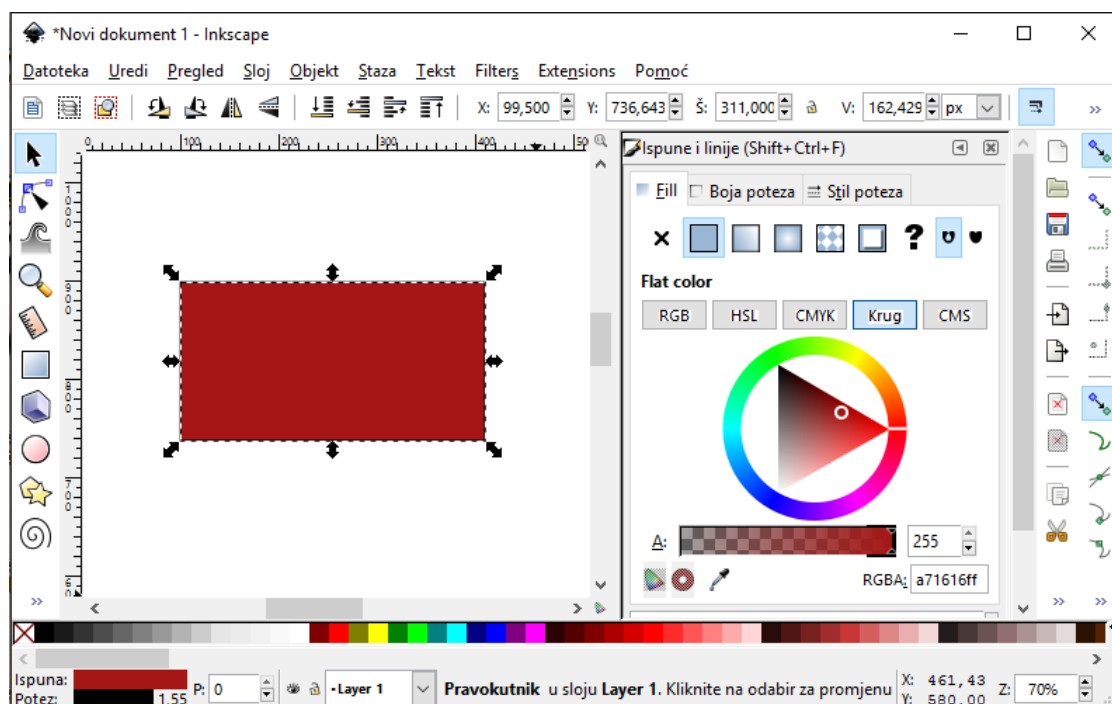
4.3. Transformacije u 3D prostoru uz pomoć CSS3 standarda

SVG sam po sebi ne omogućuje transformacije u 3D prostoru, ali pomoću CSS3 standarda takve se transformacije mogu lako napraviti. CSS3 djeluje nad elementima HTML5 standarda pa tako i nad `<svg>` elementom, te svim oznakama unutar SVG-a. Transformacije nam omogućuju skaliranje, rotiranje i translaciju elemenata preko atributa `transform`. Također, moguće je pridruživanje više transformacija jednom elementu.[2]

5. SVG editori

5.1. Inkscape

Jedan od poznatijih SVG editora je program Inkscape, a on je besplatni program za izradu i manipulaciju vektorskom grafikom. Inkscape svoje dokumente sprema u SVG formatu. Svaki nacrtani objekt u programu može se pomicati, rotirati te podešavati na razne načine. U programu se parametri transformacije oblika mogu navesti putem dijaloga "transform". Transformacije objekta se mogu podesiti na kutove, rešetke, smjernice i čvorove drugih objekata. Svaki objekt u Inkscape-u ima nekoliko atributa koji određuju svoj stil, kao što su: Fill, Stroke fill, Stroke style, opacity, filters. Inkscape ima skup alata za uređivanje putanja koji su ujedno i osnovni elementi vektorske datoteke. Jedan od glavnih prioriteta Inkscape-a je dosljednost sučelja i njegova višenamjenska upotrebljivost.[3]

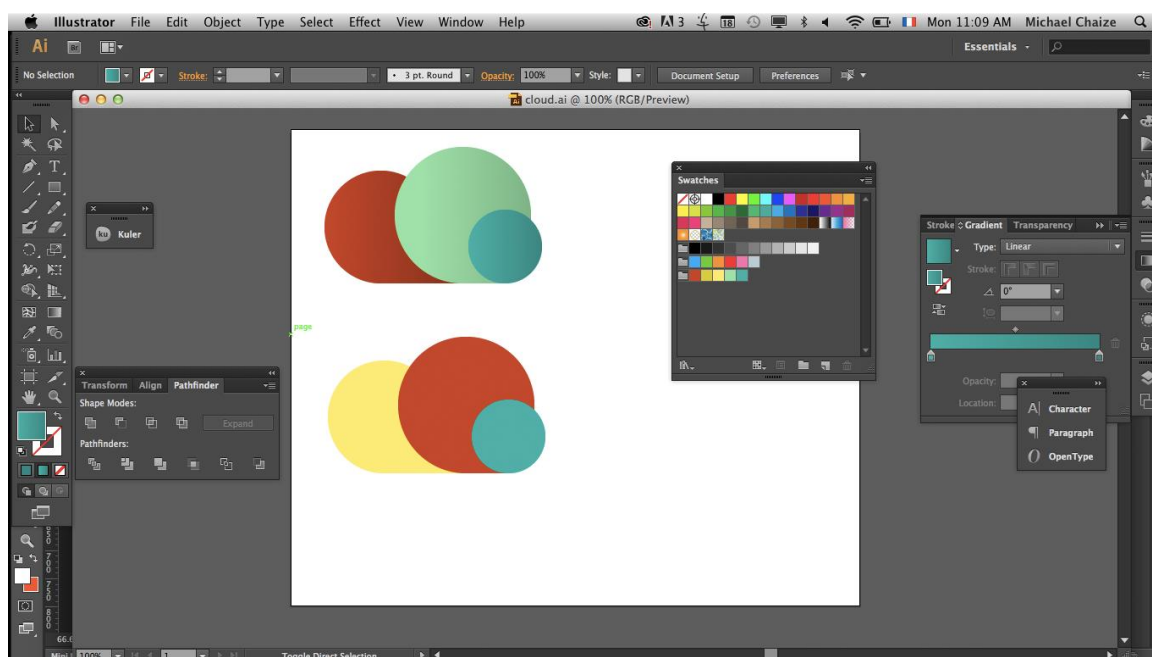


Slika 3. Prikaz kreiranja pravokutnika u Inkscape-u

Izvor: autor

5.2. Adobe Illustrator

Još od 1988.g. Illustrator je jedan od profesionalnih grafičkih alata koji je korišten od strane umjetnika, dizajnera, arhitekata i dr. Ovaj editor je jako moćan i koristi se kao standard za vektorsko crtanje. Od uključenja Adobe-a u razvoj SVG standarda, Adobe Illustrator je koristio sadržaj u SVG obliku. Zbog takvog ranog uključenja u SVG razvoj, Adobe je nedavno ponovio svoj angažman u SVG radnoj grupi.[4]



Slika 4. Adobe Illustrator

Izvor: <http://createdroplets.com/wp-content/uploads/2013/02/sc1-shapes.png>

5.3. SVG-Edit

SVG-edit je besplatan program koji omogućuje SVG crtanje. SVG-edit je dobro podržan i još uvijek u razvoju. Iako SVG-edit nema korisničku bazu i profinjenost kao Illustrator ili Inkscape, proizvodi prilično čist kod te pruža lagan način kreiranja SVG crteža. Nedostatak SVG-edita je editor Bézier krivulje jer je glomazan i ne zadovoljava korisnikova očekivanja. Unatoč navedenom nedostatku, korisnici editora ipak ga koriste jer ima jednostavno sučelje i lako konvertira crtež u SVG kodu.[4]

6. Dostupne biblioteke

Mnogo je biblioteka koje su dostupne za vektorsku grafiku, a neke od njih su: Raphaël-JavaScript Library, SVGWeb, amMap, dojox.gfx, SVGKit, PlotKit, JSDraving, VectorGraphics, i PERGOLA.[5]

6.1. Raphaël - JavaScript Library

Raphaël je mala JavaScript biblioteka koja olakšava rad s vektorskom grafikom na web-u. Raphaël koristi SVG W3C preporuku i VML kao bazu za kreiranje grafike. Svaki kreirani grafički objekt je i DOM objekt. Cilj Raphaël-a je da pruža jednostavnu upotrebu i crtanje vektorske grafike koja je kompatibilna s mnogobrojnim web preglednicima. Raphaël trenutno podržava Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+ i Internet Explorer 6.0+.[6]

6.2. SVGWeb

SVGWeb je JavaScript biblioteka koja pruža SVG podršku na mnogim web preglednicima, uključujući Internet Explorer, Firefox i Safari. SVGWeb biblioteka može biti korištena od HTML i JavaScript web stranica ili od Flash i Flex aplikacija.[7]

6.3. Dojox.gfx

Dojox.gfx (GFX) je biblioteka za vektorsku grafiku sa SVG, VML, Silvelight i Canvas podrškom. Ova biblioteka analizira sposobnosti pretraživača, detektira i istovremeno koristi najbolju metodu crtanja. Korisnik također može, ukoliko definira u postavkama, odabrati koju metodu crtanja će koristiti.[8]

6.4. SVGKit

SVGKit je kolekcija JavaScript biblioteka za klijentsku SVG manipulaciju. SVGKit pruža neovisnost pristupa web pregledniku. SVGCanvas provodi Canvas API, a SVGPlot grafikone, podatke i funkcije.[5]

6.5. amMap

AmMap je interaktivna Javascript / HTML5 biblioteka karata. Omogućuje jednostavno dodavanje interaktivnih funkcionalnih karata na web stranice i JavaScript aplikacije. Ovaj alat može se koristiti za prikaz lokacije ureda, putovanja, izradu karata itd. Fotografije ili ilustracije mogu se koristiti kao slojevi i pozadine karata.[9]

6.6. PlotKit

PlotKit je biblioteka za iscrtavanje karta i grafova. Podržan je za HTML Canvas i za SVG preko Adobe Viewer-a i podrške pretraživača.[5]

6.7. JSDrawing

JSDrawing je JavaScript biblioteka koja omogućuje korištenje vektorske grafike, neovisno o pregledniku i platformi. API je baziran na SVG komandama, što omogućava jednostavno korištenje.[5]

6.8. VectorGraphics

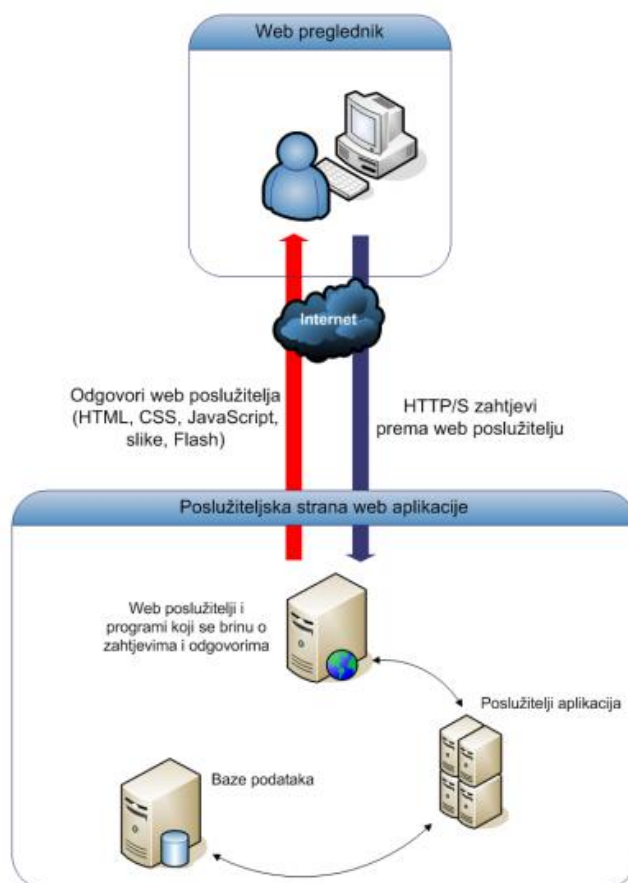
Ova JavaScript biblioteka omogućava grafičke mogućnosti za JavaScript funkcije da crta krugove, elipse, kose linije i poligone dinamički u web stranicu. Korištenje ove vektorske grafičke biblioteke ne zahtjeva mnogo prethodnog iskustva u radu s JavaScriptom.[5]

6.9. PERGOLA

PERGOLA je JavaScript biblioteka koja koristi SVG, dizajnirana je da omogući programerima moćan alat za izgradnju web aplikacija, korisničkih sučelja, prezentacija...[5]

7. Web aplikacije

U ranijim načinima komunikacije između klijenta i poslužitelja, svaka aplikacija je imala svoj klijentski program koji je služio kao korisničko sučelje. Promjena aplikacije zahtijevala je i promjenu upotrebe iste od strane klijenata, a to je povećalo troškove podrške i ujedno smanjilo produktivnost. Za svaki operacijski sustav klijentska aplikacija je morala biti drugačija. Korisnicima je bilo onemogućeno ili otežano korištenje aplikacija jer su se proizvođači odlučili samo za jednu tehnologiju i jedan operacijski sustav. S druge strane web aplikacije se strukturiraju kao niz HTML dokumenata pa svatko može koristiti web aplikaciju bez instaliranja posebnih aplikacija jer HTML tehnologija nije vezana uz operacijski sustav i na taj način je pojednostavljeno pokretanje, podrška i poboljšana aplikacija. Jedino čega se razvojni inženjeri moraju pridržavati su propisani standardi tehnologija, jer im oni postaju nova razvojna platforma, a neke od njih su: XML, CSS, HTML i JavaScript.[1]



Slika 5. Dijagram komunikacije između korisnika i web aplikacije

Izvor: <http://www.zemris.fer.hr/predmeti/rg/diplomski/08Rebrovic/webAplikacije.html>

8. Priprema za rad sa SVG grafikom

Za gledanje SVG sadržaja potreban je bilo koji moderan pretraživač. Na primjer, Internet Explorer, Firefox, Chrome, Safari, Opera podržavaju gledanje SVG sadržaja. Za sve navedene pretraživače najbolje je koristiti najnovije verzije za bolju podršku SVG-a.

Primjer:

```
<?xml version="1.0" standalone="yes"?>
<studentskiRadovi>
<rad tip="Završni rad">
<student>Franjo Galeković</student>
<naziv>SVG grafike na web-u</naziv>
</rad>
</studentskiRadovi>
```

Redak `<?xml version="1.0" standalone="yes"?>` jest opcionalna direktiva koja govori da se radi o XML verziji 1.0 i kada ne bi bilo dodatnih dokumenata, bez kojih bi prikaz dokumenta bio nemoguć.[10]

Primjer:

```
<svg xmlns="http://www.w3.org/2000/svg">
<circle r="50" cx="50%" cy="50%"/>
</svg>
```

Svaka informacija o SVG crtežu počinje pojavom `<svg>` elementa. Takav element sadrži i definicije o dosegima imena (engl. *Namespace*). Nužan dio svakog SVG dokumenta jest redak `<svg xmlns="http://www.w3.org/2000/svg"></svg>`. Unutar prvog `<svg>` elementa, koji je i korijenski element XML dokumenta nalaze se ostale informacije.[10]

9. Izrada osnovnih vektorskih oblika

U primjerima koji slijede prikazani su primjeri izrade: linije, pravokutnika, kruga, elipse, poligona, izlomljene linije, te korištenje putanje i izrada uzorka pomoću pravokutnika.

9.1 Linije

Za stvaranje vidljive linije, potrebno je postaviti x1 i y1 za početnu točku, te x2 i y2 za drugu točku vrijednosti <line> elementa. Potrebno je dodatno postaviti boju linije i druga svojstva kako bi linija bila vidljiva.

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2   <line x1="100" y1="100" x2="300" y2="100" stroke="green"
3     stroke-width="10" stroke-linecap="round" />
4 </svg>
```

Slika 6. Linija

Izvor: autor

Atributi:

Stroke: određuje boju linije

Stroke-width: određuje se debljina linije

Stroke-linecap: definira se oblik na krajevima linije



Slika 7. Linija

Izvor: autor

9.2. Pravokutnik

Pravokutni element `<rect>` zahtijeva attribute širine i visine, a mogu se odrediti `x` i `y` atributi koji određuju poziciju u odnosu na gornji lijevi kut SVG canvasa. Dodatni `rx` i `ry` atributi određuju zaobljivanje uglova.

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2   <rect x="50" y="50" width="300" height="170" rx="90" ry="50"
3     stroke="darkseagreen" stroke-width="10"
4     fill="lightgray" fill-opacity="0.6" />
5 </svg>
```

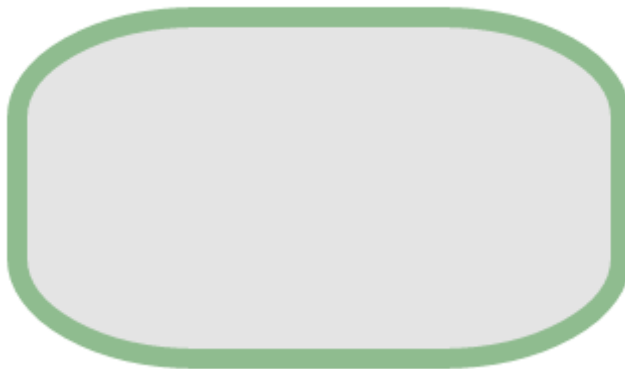
Slika 8. Pravokutnik

Izvor: autor

Atributi:

Fill: određuje boju oblika

Fill-opacity: prozirnost oblika



Slika 9. Pravokutnik

Izvor: autor

9.3. Krug i elipsa

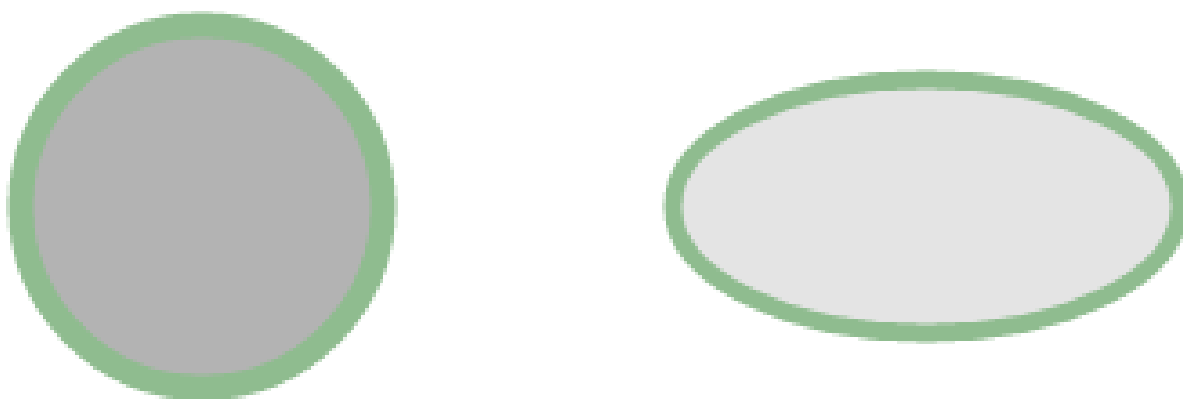
SVG <circle> element zahtijeva postavljanje radijusa r kako bi se odredila veličina radijusa nacrtanog kruga. Elementu <ellipse> potrebno je dodati atribute rx i ry za određivanje radijusa x i y vrijednosti nacrtane elipse.

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2   <circle cx="150" cy="150" r="50"
3     stroke="darkseagreen" stroke-width="7"
4     fill="grey" fill-opacity="0.6"/>
5   <ellipse cx="350" cy="150" rx="70" ry="35"
6     stroke="darkseagreen" stroke-width="5"
7     fill="lightgray" fill-opacity="0.6" />
8 </svg>
```

Slika 10. Krug i elipsa

Izvor: autor



Slika 11. Krug i elipsa

Izvor: autor

9.4. Izlomljene linije i poligoni (engl. *Polylines and Polygons*)

Izlomljene linije i poligoni su slični jedan drugom te oboje zahtijevaju atribute točaka, koji se sastoje od x,y parova vrijednosti. Ova dva oblika pružaju crtanje serija ravnih linija. Glavna razlika između <polyline> i <polygon> oblika u SVG-u je ta da put izlomljenih linija neće biti zatvoren.

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2   <polygon points="100,50 115,120 150,150
3     115,180 100,250 85,180 50,150 85,120"
4     fill="darkorange" fill-opacity="0.5"
5     stroke="papayawhip" stroke-width="20"
6     stroke-opacity="0.7" stroke-linejoin="miter"/>
7   <polyline points="200,60 240,230 310,230 350,60"
8     fill="lightcyan" fill-opacity="0.7"
9     stroke="darkviolet"
10    stroke-width="25" stroke-linecap="round"
11    stroke-opacity="0.2" />
12 </svg>
```

Slika 12. Poligon i izlomljene linije

Izvor: autor



Slika 13. Poligon i izlomljene linije

Izvor: autor

9.5. Putanja (engl. *Path*)

SVG `<path>` element koristi se za crtanje putanji koje tipično započinju M podnaredbom, a ona upućuje na početak crteža (x,y).

Primjer:

```
d= "M 100 100 L 150 150..."
```

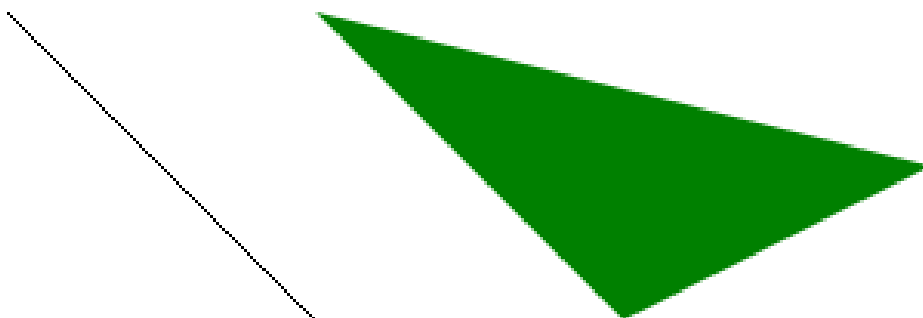
Opcije crtanja od točke do točke su: linije (L), kvadratične krivulje (Q), kockaste krivulje (C), ili luk (A).

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2   <path stroke="black"
3     d="M 50 50 L 150 150"/>
4   <path d="M 150 50
5     L 250 150 350 100" fill="green"/>
6 </svg>
```

Slika 14. Putanja

Izvor: autor



Slika 15. Putanja

Izvor: autor

9.6. Uzorci (engl. *Patterns*)

Upotrebom SVG jezika lako se kreiraju i koriste uzorci. U primjeru će se prikazati uzorak sa 10x10 pixela pravokutnika. <defs> element se može iskoristiti za pohranu sadržaja koji neće biti direktno prikazan. Takav pohranjen sadržaj se može uputiti i biti prikazan s drugim SVG elementima.

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg">
2 <defs>
3   <pattern id="Uzorak" width="10" height="10"
4   <patternUnits="userSpaceOnUse">
5     <rect width="10" height="10" fill="lightcyan"
6     stroke="green" stroke-width="0.2"/>
7   </pattern>
8 </defs>
9 <rect id="Pozadina" x="0" y="0" width="100%"
10 height="100%" fill="url(#Uzorak)" stroke-width="0.5"
11 stroke="#000000" />
12 </svg>
```

Slika 16. Uzorak

Izvor: autor



Slika 17. Uzorak

Izvor: autor

10. Dodavanje teksta, stila i transformacije

10.1 Dodavanje i pozicioniranje teksta

Dodavanje teksta je prilično jednostavno. Tekst može mijenjati stil, oblik i razmak na puno načina. Neki od najvažnijih atributa `<text>` elementa su `x` (horizontalna pozicija), `y` (vertikalna pozicija), `fill`, `font-size` i `font-family`.

Primjer korištenja:

```

1  <svg xmlns="http://www.w3.org/2000/svg"
2  xmlns:xlink="http://www.w3.org/1999/xlink">
3  <defs>
4  <path id="put" d="M 10 100 C 200 30 300 250 350 50 400 200" />
5  </defs>
6  <text font-family="serif" font-style="oblique"
7  text-decoration="underline" font-size="17"
8  fill="purple" font-weight="bold" >
9  <textPath xlink:href="#put">Ovo je primjer teksta
10 koji se giba po putanji.</textPath>
11 </text>
12 </svg>

```

Slika 18. Gibanje teksta po putanji

Izvor: autor

Atributi:

`xlink:href`: definira link na izvor

`Font-family`: tema teksta

`Font-style`: određuje stil teksta

`Text-decoration`: dekoracija teksta

`Font-size`: veličina teksta

`Font-weight`: debljina teksta

Ovo je primjer teksta koji se giba po putanji.

Slika 19. Gibanje teksta po putanji

Izvor: autor

10.2 Gradijenti

Gradijenti u grafičkom svijetu znače postupni prijelaz u različitim smjerovima iz jedne boje u drugu. Offset atribut se označava brojem ili postotkom i označava gdje završava gradijent. "Stop-collor" označava koja boja se koristi na stajalištu gradijenta. "Stop-opacity" označuje prozirnost danog stajališta gradijenta.

Primjer:

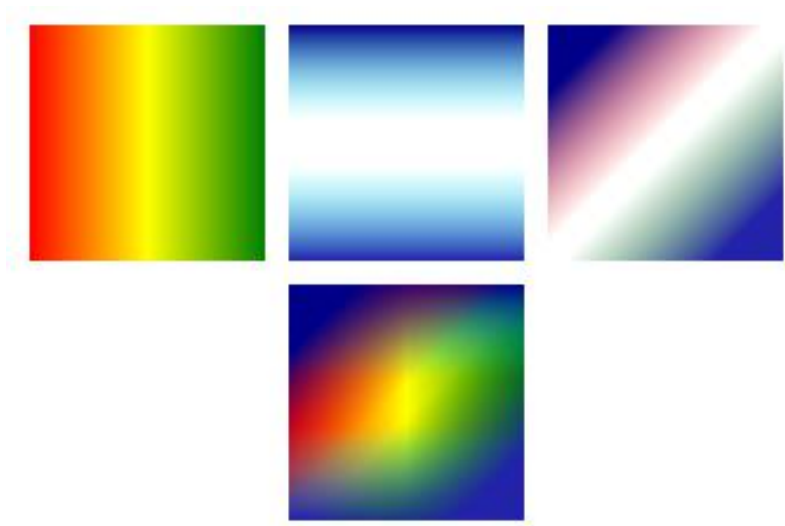
```
<stop offset="1" stop-collor="red" stop-opacity="0"/>
```

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg"
2   xmlns:xlink="http://www.w3.org/1999/xlink">
3   <linearGradient id="r" >
4     <stop offset="0" stop-color="red"/>
5     <stop offset=".5" stop-color="yellow"/>
6     <stop offset="1" stop-color="green"/>
7   </linearGradient>
8   <linearGradient id="t" gradientTransform="rotate(90,.5,.5)">
9     <stop offset="0" stop-color="#008" />
10    <stop offset=".4" stop-color="cyan" stop-opacity="0"/>
11    <stop offset=".6" stop-color="cyan" stop-opacity="0"/>
12    <stop offset="1" stop-color="#22a" />
13  </linearGradient>
14  <linearGradient id="s" gradientTransform="rotate(45,.5,.5)">
15    <stop offset="0" stop-color="#008" />
16    <stop offset=".45" stop-color="red" stop-opacity="0"/>
17    <stop offset=".55" stop-color="green" stop-opacity="0"/>
18    <stop offset="1" stop-color="#22a" />
19  </linearGradient>
20  <rect x="10" y="10" height="100" width="100" fill="url(#r)"/>
21  <rect x="120" y="10" height="100" width="100" fill="url(#t)"/>
22  <rect x="230" y="10" height="100" width="100" fill="url(#s)"/>
23  <rect x="120" y="120" height="100" width="100" fill="url(#r)"/>
24  <rect x="120" y="120" height="100" width="100" fill="url(#t)"/>
25  <rect x="120" y="120" height="100" width="100" fill="url(#s)"/>
26 </svg>
```

Slika 20. Primjena gradijenata na pravokutnike

Izvor: autor



Slika 21. Primjena gradijenata na pravokutnike

Izvor: autor

10.3. Isječci

`<clipPath>` element daje mogućnost definiranja oblika. Clip path se može primijeniti na bilo koji nacrtani objekt, uključujući i grupe objekata. Također, clip path se sam može sastojati od mnogih oblika. Za primjer isječka uzeta je slika 21. koja se prikazuje kao isječak u obliku elipse na slici 23.

Primjer korištenja:

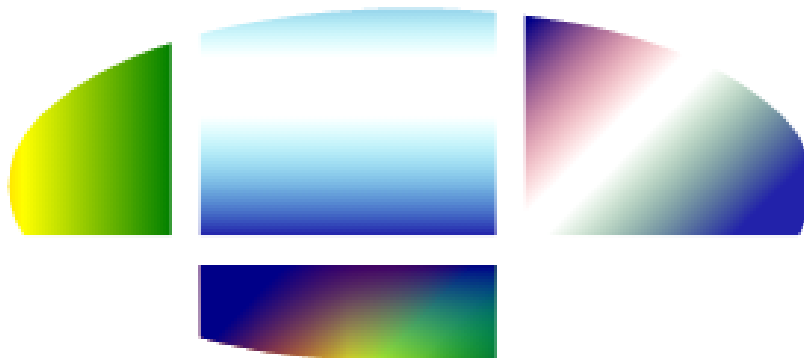
```

1  <svg xmlns="http://www.w3.org/2000/svg"
2  <xmlns:xlink="http://www.w3.org/1999/xlink">
3  <clipPath id="CP">
4  <ellipse cx="29%" cy="26%" rx="10%" ry="9%"/>
5  </clipPath>
6  <image y="12%" x="15%" width="40%" height="55%"
7  xlink:href='gradijenti.svg' clip-path="url(#CP)"/>
8  </svg>

```

Slika 22. Isječak u obliku elipse

Izvor: autor



Slika 23. Isječak u obliku elipse

Izvor: autor

10.4. Maskiranje

U SVG-u bilo koji grafički objekt se može koristiti za maskiranje. Za primjer maskiranja prikazan je tekst na kojem se maskiraju prva dva slova teksta. Maska je napravljena u obliku pravokutnika, koji je ispunjen gradijentom.

Primjer:

```
<mask id="Ma">  
  
<rect x="300" y="300" width="400" height="100"  
fill="url(#gradijent)"/>  
  
</mask>
```

Primjer korištenja:

```

1  <svg xmlns="http://www.w3.org/2000/svg"
2  xmlns:xlink="http://www.w3.org/1999/xlink">
3  <linearGradient id="gradijent" >
4  <stop offset="0.0" stop-color="white"/>
5  <stop offset="1" stop-color="black"/>
6  </linearGradient>
7  <mask id="Ma">
8  <rect x="300" y="300" width="400" height="100"
9  fill="url(#gradijent)"/>
10 </mask>
11 <text x="220" y="365" font-family="impact"
12 font-size="52" mask="url(#Ma)"
13 fill="black">Maskiranje teksta.</text>
14 </svg>

```

Slika 24. Maskiranje teksta

Izvor: autor

Maskiranje teksta.

Slika 25. Maskiranje teksta

Izvor: autor

10.5. Transformacije

SVG jezik pruža korisne metode repositioniranja individualnih objekata ili grupa objekata pomoću jednostavnih naredbi za transformiranje: `translate`, `scale`, `rotate`, `skewX`, `skewY`.

Naredba `translate` omogućuje repositioniranje sadržaja određivanjem `xy`-koordinata u ovom formatu: `transform="translate(50,10)"`, naredba `scale` omogućuje skaliranje elementa na koji je primijenjen s obzirom na koordinatni sustav. Koristeći `skewX` i `skewY` naredbu oblici se mogu iskrivljavati horizontalno i vertikalno.

Primjer korištenja:

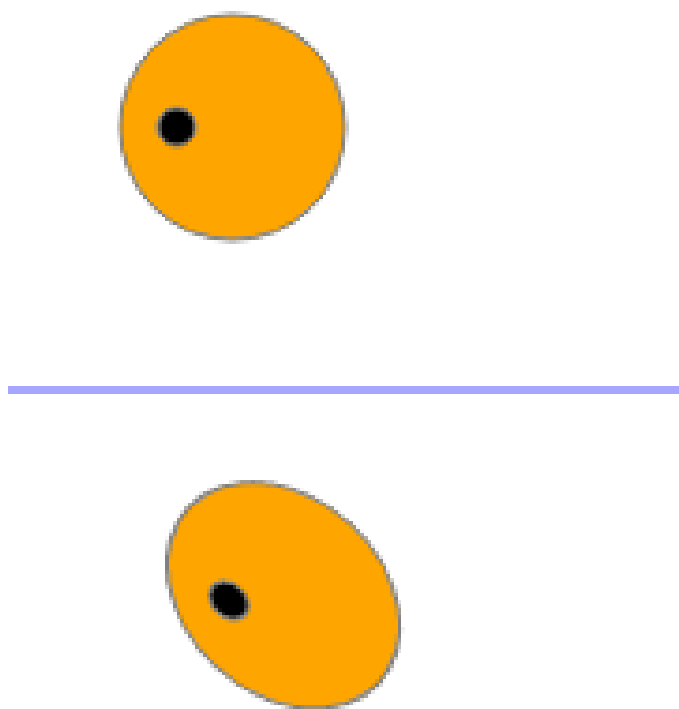
```

1  <svg xmlns="http://www.w3.org/2000/svg"
2  <xmlns:xlink="http://www.w3.org/1999/xlink">
3  <line id="voda" x1="-50" y1="110" x2="100%" y2="110"
4  stroke="blue" stroke-width="1" stroke-opacity="0.7" />
5  <g id="scena">
6  <circle id="sunce" r="30" cx="60" cy="40"
7  fill="orange" stroke="grey" stroke-width="1" />
8  <circle id="tranzicija" r="5" cx="45" cy="40"
9  fill="black" stroke="grey" stroke-width="1" />
10 </g>
11 <use xlink:href="#scena"
12 transform="scale(1 -1) translate(30 -210)
13 skewX(-20) skewY(5)" />
14 </svg>

```

Slika 26. Transformacija

Izvor: autor

**Slika 27.** Transformacija

Izvor: autor

11. Animacije

Iako SVG može biti statički format slike pogodan za ilustracije, još od osnutka je dizajniran da bude interaktivan i dinamičan. Animacija započinje naredbom *"animate"*.

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg"
2   xmlns:xlink="http://www.w3.org/1999/xlink"
3   width="100%" height="100%">
4   <rect x="100" y="85" rx="12" height="30"
5     width="150" fill="purple" stroke="black"
6     stroke-width="3" >
7     <animate attributeName="width" dur="3s"
8       values="150;100;150" repeatCount="indefinite"/>
9   </rect>
10  <ellipse id="E" cx="100" cy="100" rx="30" ry="40"
11    fill="#448" opacity=".75" stroke="black"
12    stroke-width="6" stroke-dasharray="8,4">
13    <animate attributeName="rx" dur="3s" values="10;70;10"
14      repeatCount="indefinite"/>
15    <animate attributeName="ry" dur="5s" values="30;60;30"
16      repeatCount="indefinite"/>
17  </ellipse>
18  <ellipse cx="250" cy="100" rx="30" ry="40" fill="#448"
19    opacity=".75" stroke="black"
20    stroke-width="6" stroke-dasharray="8,4">
21    <animate attributeName="rx" dur="5s" values="70;10;70"
22      repeatCount="indefinite"/>
23    <animate attributeName="ry" dur="3s" values="60;30;60"
24      repeatCount="indefinite"/>
25    <animate attributeName="cx" dur="3s" values="250;200;250"
26      repeatCount="indefinite"/>
27  </ellipse>
28 </svg>
```

Slika 28. Animacija

Izvor: autor

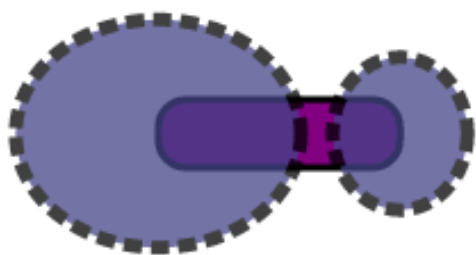
Atributi:

attributeName: odabire atribut objekta za animaciju

dur: dužina odvijanja animacije

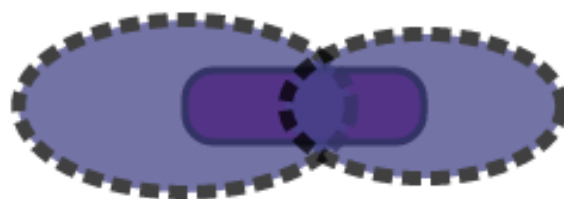
values: početne i završne vrijednosti animacije

repeatCount: broj ponavljanja animacije



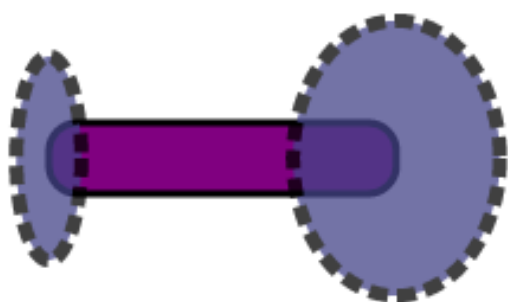
Slika 29. Animacija 01

Izvor: autor



Slika 30. Animacija 02

Izvor: autor



Slika 31. Animacija 03

Izvor: autor

11.1. Interakcija s animacijom

SVG sadržaj može biti responsivan na akcije korisnika. `<animateMotion>` se koristi za animaciju objekata koji se kreću po putanji.

Primjer korištenja: `<animate>` element sadrži atribut `begin="G.click"`. Animacija počinje u onom trenutku kad se klikne objekt elipse `id="G"`.

```

1  <svg xmlns="http://www.w3.org/2000/svg"
2  xmlns:xlink="http://www.w3.org/1999/xlink">
3  <path id="krivina" stroke="black" fill="none"
4  stroke-width="3" fill-opacity=".5"
5  d="M 0,100 C 100,150 100,50 200,50
6  C 300,50 300,150 400,100" />
7  <ellipse cx="0" cy="0" rx="16" ry="8"
8  fill="green" opacity=".85" stroke="black"
9  stroke-width="2">
10 <animateMotion dur="3s" rotate="auto"
11 repeatCount="2" begin="G.click">
12 <mpath xlink:href="#krivina"/>
13 </animateMotion>
14 </ellipse>
15 <g id="G">
16 <ellipse cx="200" cy="90" rx="33" ry="15"
17 fill="yellow" stroke="black" stroke-width="2" />
18 <text x="175" y="101" font-size="31"
19 fill="black" font-family="arial">GO</text>
20 </g>
21 </svg>

```

Slika 32. Interakcija s animacijom

Izvor: autor



Slika 33. Interakcija s animacijom

Izvor: autor

11.2. Skriptiranje

JavaScript ugrađen u SVG obično prebiva unutar oznake `<script>`. Počinje oznakom `<script><![CDATA[, a završava oznakom]]></script>`.

Primjer korištenja: Svakom novom gumbu je dana nova lokacija, a ostale attribute nasljeđuje od prethodnika. Ako se klikne prvi gumb, novi gumb dobiva novi broj i nasumičnu boju. Ako se klikne naknadni gumb, novom gumbu se pridaje dio imena i boja tog kliknutog gumba.

```

1  <svg xmlns="http://www.w3.org/2000/svg" width="100%"
2  xmlns:xlink="http://www.w3.org/1999/xlink">
3  <script><![CDATA[ xmlns = "http://www.w3.org/2000/svg";
4  xlink = "http://www.w3.org/1999/xlink";
5  var count = 0;
6  var letters = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J"];
7  var letco = 0;
8  function add (evt) {
9    var C = evt.currentTarget;
10   var N = C.cloneNode(true);
11   var x = Math.random() * 700 + 5;
12   var y = Math.random() * 400 + 5;
13   N.setAttributeNS(null, "transform", "translate(" + x + "," + y + ")");
14   var TspansN = N.getElementsByTagNameNS(xmlns, "tspan");
15   var TspansC = C.getElementsByTagNameNS(xmlns, "tspan");
16   if (TspansC.item(1).textContent == "Me") {
17     N.setAttributeNS(null, "fill", Color());
18     TspansN.item(1).textContent = letters[letco ++% letters.length];
19   } else {
20     N.setAttributeNS(null, "fill", C.getAttributeNS(null, "fill"));
21     TspansN.item(1).textContent = TspansN.item(1).textContent;
22   }
23   TspansN.item(2).textContent = count++;
24   document.documentElement.appendChild(N);
25   } function Color () {
26     return "rgb(" + parseInt(Math.random() * 255) + ","
27     + parseInt(Math.random() * 255) + ","
28     + parseInt(Math.random() * 255) + ")";
29   } ]]></script>
30   <g onmousedown="add(evt)" fill="orange"
31   transform="translate(300,200)">
32   <rect height="30" width="80" fill="inherit" rx="10"
33   stroke="black" stroke-width="2"
34   fill-opacity=".7" /><text font-size="17" font-family="serif"
35   transform="translate(6,20)" fill="black">
36   <tspan>Click </tspan><tspan>Me</tspan><tspan></tspan>
37   </text></g></svg>

```

Slika 34. Dodavanje elemenata

Izvor: autor

Metode:

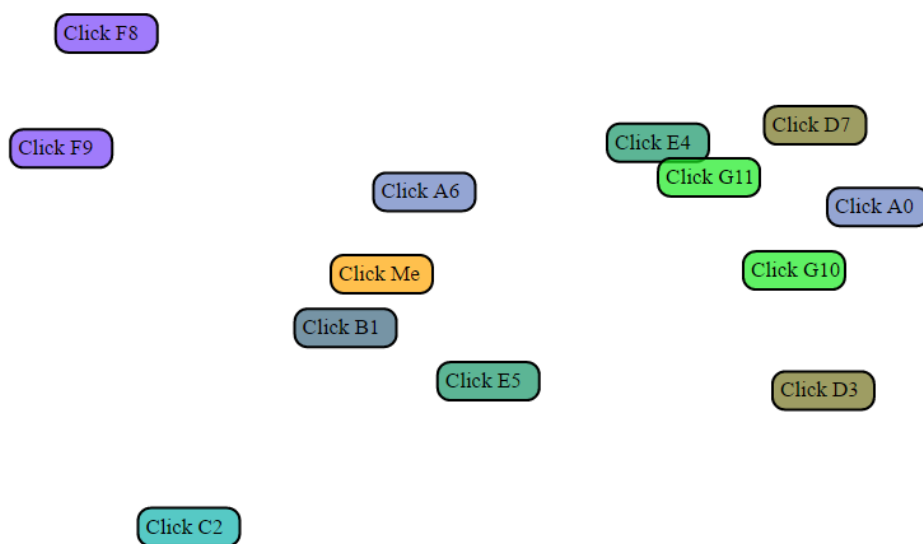
`cloneNode()`: klonira sve atribute i vrijednosti

`setAttributeNS()`: dodaje novi atribut ili mijenja vrijednost atributa

`getElementsByTagNameNS()`: vraća listu elemenata s danom oznakom

`getAttributeNS()`: vraća string vrijednost atributa s određenim imenom

`appendChild()`: dodaje čvor na kraj liste



Slika 35. Dodavanje elemenata

Izvor: autor

Primjer korištenja: Svaki klonirani krug u sebi sadrži funkciju "remove". Funkcija "remove" koristi event da odredi koji objekt je odabran.

```

1  <svg xmlns="http://www.w3.org/2000/svg" width="100%"
2  xmlns:xlink="http://www.w3.org/1999/xlink" >
3  <script><![CDATA[
4      xmlns = "http://www.w3.org/2000/svg";
5      xlink = "http://www.w3.org/1999/xlink";
6      var lastOne;
7      Root=document.documentElement;
8      function Color () {
9          var R = parseInt(Math.random() * 255);
10         var G = parseInt(Math.random() * 255);
11         var B = parseInt(Math.random() * 255);
12         return "rgb(" + R + "," + G + "," + B + ")";
13     }
14     function add (evt) {
15         var C = document.getElementById("C");
16         var NC = C.cloneNode(false);
17         NC.setAttributeNS(null, "cx", evt.clientX);
18         NC.setAttributeNS(null, "cy", evt.clientY);
19         NC.setAttributeNS(null, "fill", Color()); NC.removeAttribute("id");
20         Root.appendChild(NC);
21     }
22     function remove (evt) { Root.removeChild(evt.target); }
23     ]]></script>
24     <defs>
25         <circle r="20" fill-opacity=".5" id="C" onclick="remove(evt)"
26         stroke="black" stroke-width="2" stroke-dasharray="8,4" />
27     </defs>
28     <rect width="100%" height="100%" fill="white" onclick="add(evt)" />
29     <rect x="25" y="30" height="50" width="235" stroke="#800"
30     stroke-width="2" fill="grey" opacity=".65">
31     </rect>
32     <text x="37" y="50" font-size="12" font-family="arial" fill="#800">
33     Kliknite na prazno mjesto za dodavanje</text>
34     <text x="59" y="70" font-size="12" font-family="arial" fill="#800">
35     Za brisanje odaberite krug</text>
36     </svg>

```

Slika 36. Dodavanje i brisanje elemenata

Izvor: autor

Metode:

getElementById: upućuje na element prema Id-u

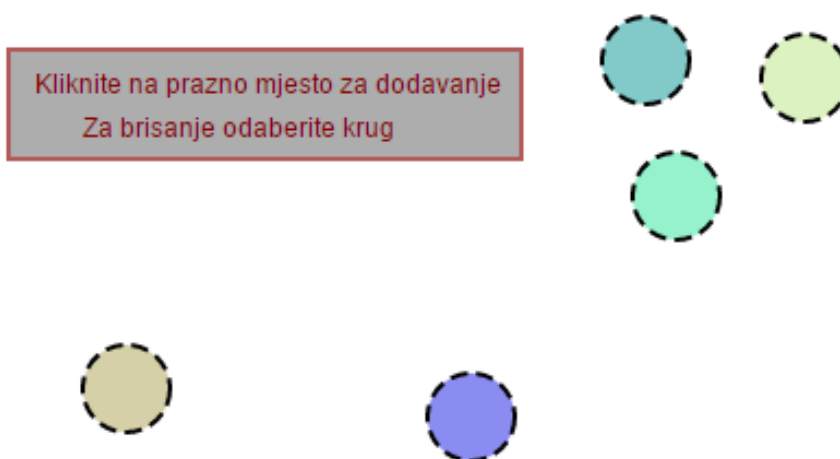
removeAttribute: uklanja atribut elementa

removeChild: uklanja čvor child iz DOM-a



Slika 37. Dodavanje i brisanje elemenata 01

Izvor: autor



Slika 38. Dodavanje i brisanje elemenata 02

Izvor: autor

12. Filteri

Filteri se koriste za dodavanje specijalnih efekata SVG grafike. Za definiranje filtera koristi se <filter> element.

Primjer korištenja:

```
1 <svg xmlns="http://www.w3.org/2000/svg"
2   xmlns:xlink="http://www.w3.org/1999/xlink" >
3   <filter id="A" color-interpolation-filters="sRGB">
4     <feComponentTransfer>
5       <feFuncR type="linear" slope="-1" intercept="1" />
6       <feFuncG type="linear" slope="-1" intercept="1" />
7       <feFuncB type="linear" slope="-1" intercept="1" />
8     </feComponentTransfer>
9   </filter>
10  <filter id="B">
11    <feGaussianBlur stdDeviation="5"/>
12  </filter>
13  <filter id="C">
14    <feColorMatrix type="saturate" values="0" />
15  </filter>
16  <image id="S" x="0" y="0" width="200" height="200"
17    preserveAspectRatio="none" xlink:href="logo.png" />
18  <use xlink:href="#S" filter="url(#A)" transform="translate(200,0)" />
19  <use xlink:href="#S" filter="url(#B)" transform="translate(400,0)" />
20  <use xlink:href="#S" filter="url(#C)" transform="translate(600,0)" />
21 </svg>
```

Slika 39. Primjena filtera

Izvor: autor

Naredbe:

feGaussianBlur: koristi se za zamagljivanje

feColorMatrix: mijenja boju na osnovi matrix transformacije

feComponentTransfer: omogućuje neovisnu manipulaciju kanala svake boje.



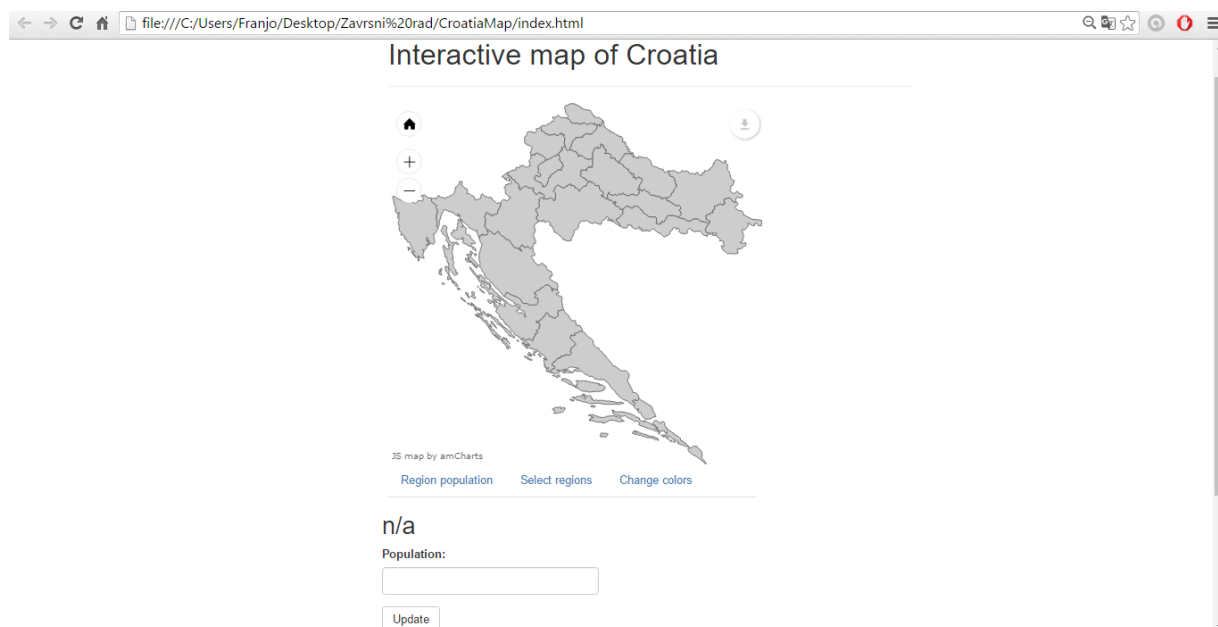
Slika 40. Primjena filtera

Izvor: autor

13. Interaktivna SVG karta Hrvatske

U ovome radu izrađena je interaktivna SVG karta. Radi se o jednostavnoj aplikaciji u kojoj je moguće odabrati županije Hrvatske te kao takve odabrane spremiti u png, jpg, svg i pdf formatu. Ova karta može služiti kao smjernica za posjećene dijelove Hrvatske ili u neke edukativne svrhe.

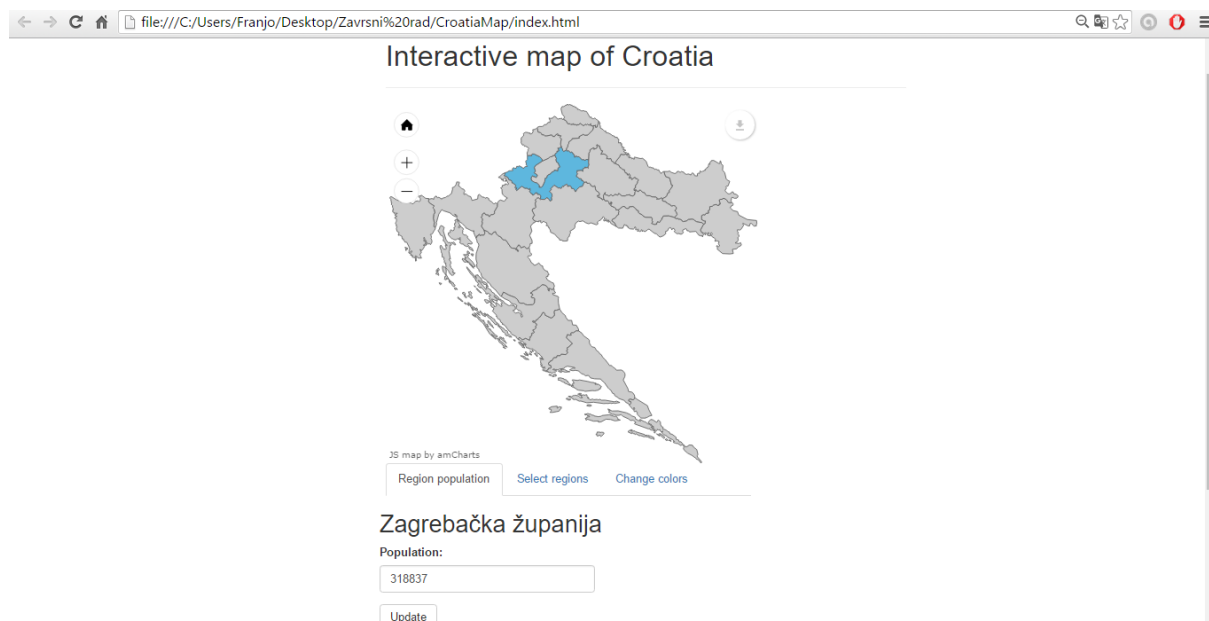
Indeks stranica sadrži opcije poput: povećavanje i smanjivanje karte, gumb stanovništvo regije (Region population) gdje je moguće vidjeti i mijenjati broj stanovnika županije, gumb za odabrane regije (Select regions), promjena boje (change collors), te gumb za preuzimanje, bilješke i printanje.



Slika 41. Indeks stranica

Izvor: autor

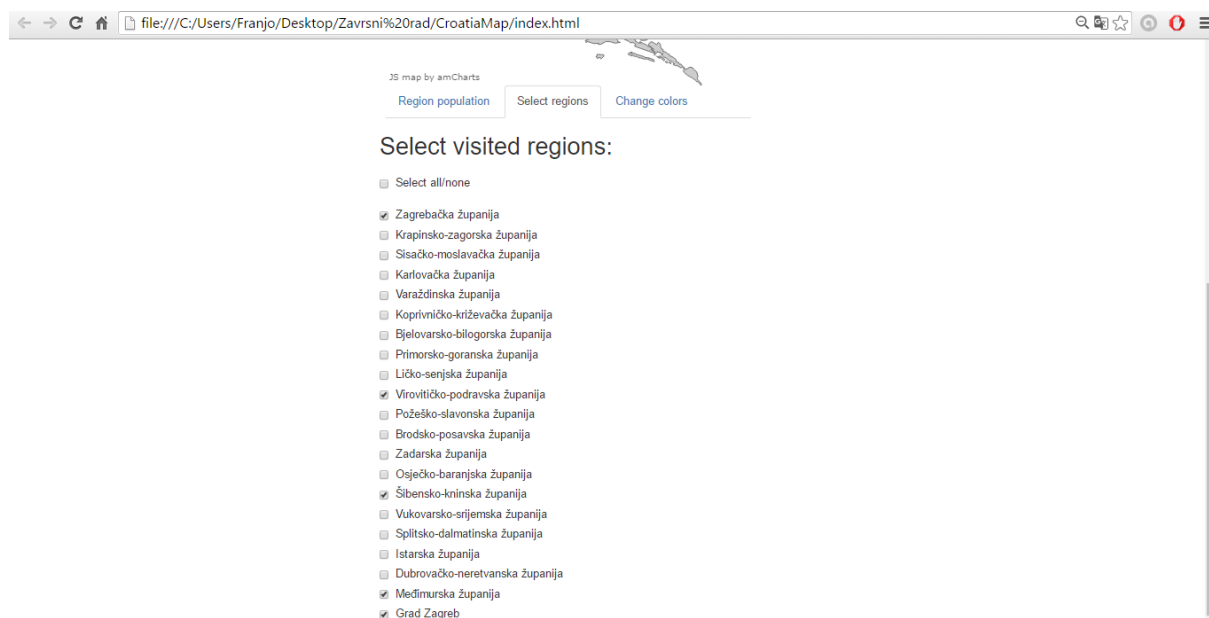
Prikaz broja stanovnika određene županije prikazuje se klikom na gumb "Region population" i klikom na županiju. Napravljen je proces koji komunicira sa MySQL bazom pa se broj stanovnika županije može izmijeniti od strane korisnika aplikacije.



Slika 42. Populacija regije

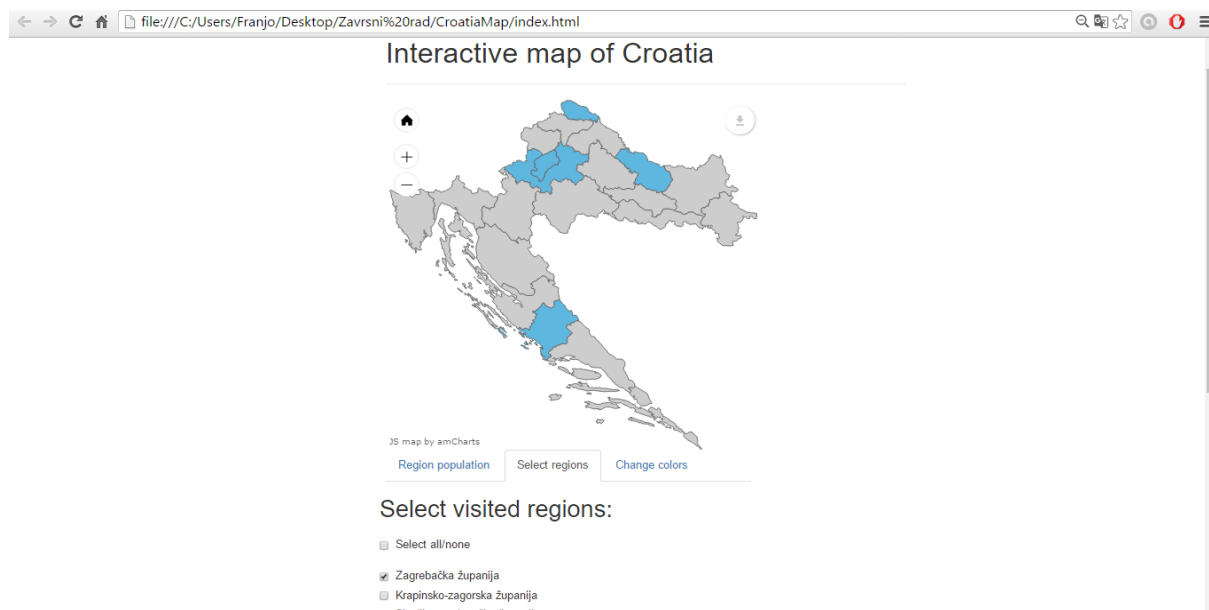
Izvor: autor

Popis svih županija i onih odabranih, prikazuje se klikom na "Select regions". Ovdje se mogu odabrati i poništavati odabrane županije, što i vizualno utječe na samu kartu.



Slika 43. Selektiranje regija 01

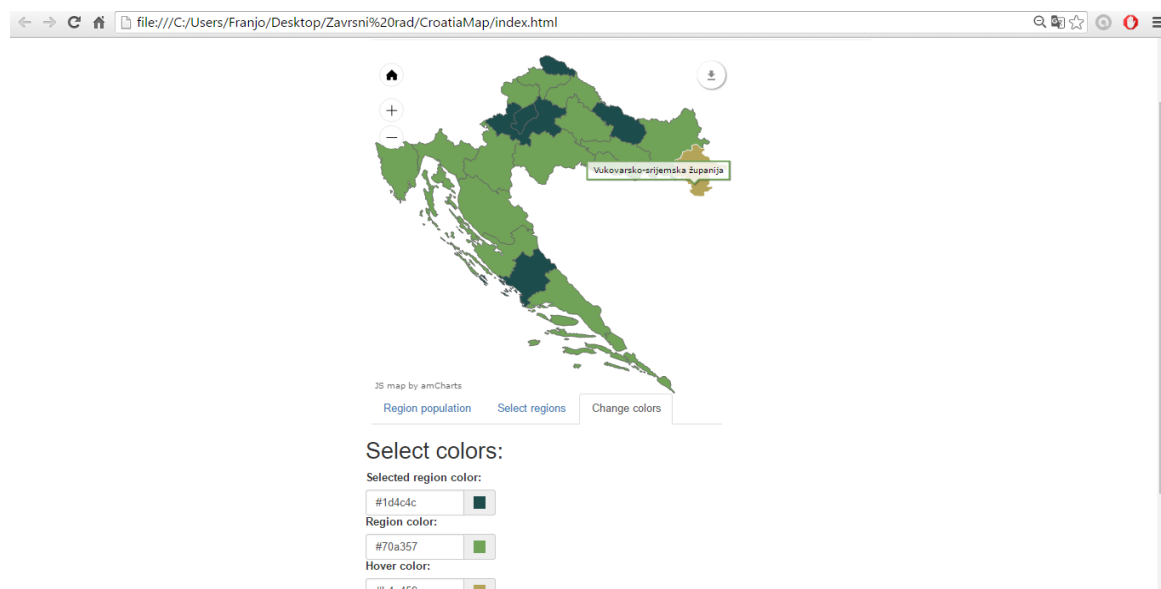
Izvor: autor



Slika 44. Selektiranje regija 02

Izvor: autor

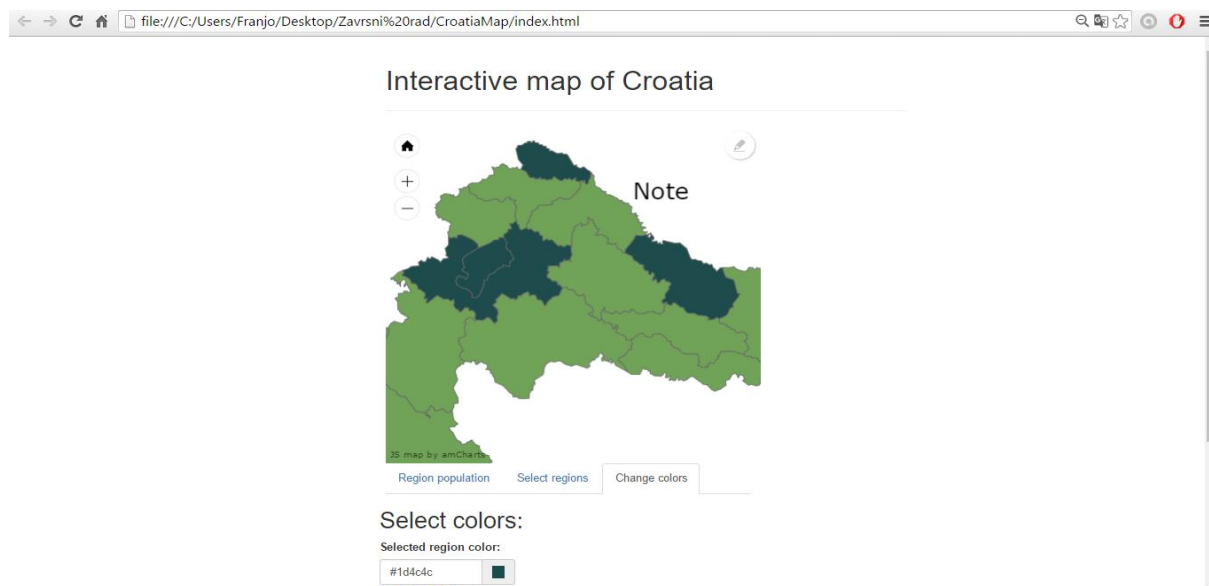
Klikom na gumb "Change color" pojave se opcije za: promjenu boja odabranih regija, boja regije i boje regije prilikom selektiranja.



Slika 45. Promjena boje

Izvor: autor

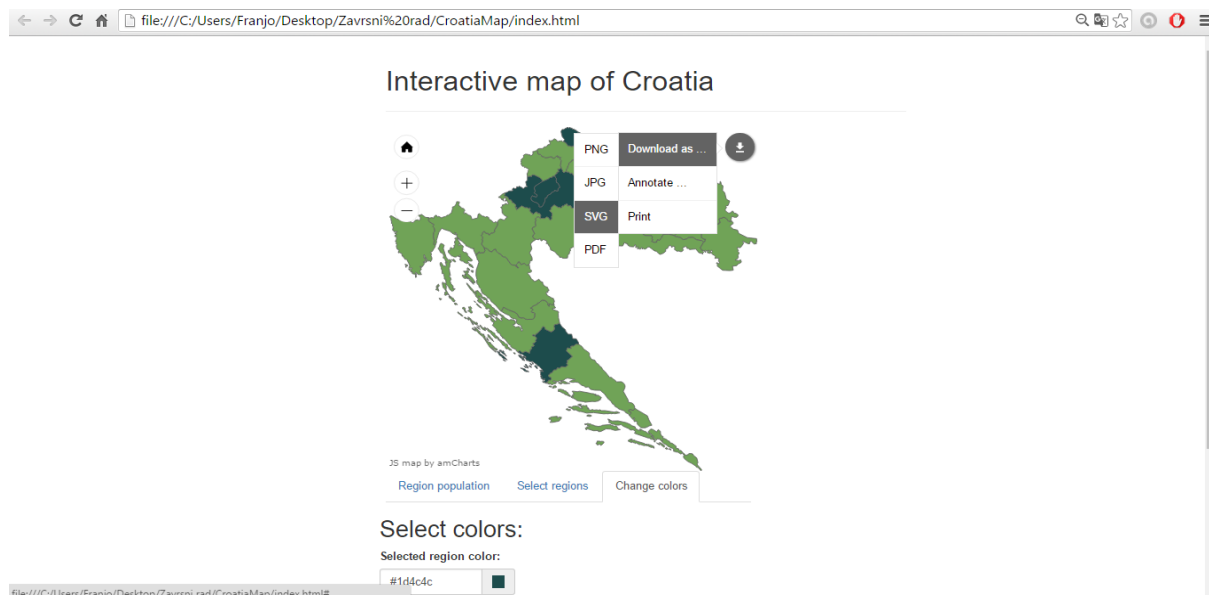
Karta ima opcije povećavanja, smanjivanja i izradu bilješki.



Slika 46. Povećavanje karte i izrada bilješke

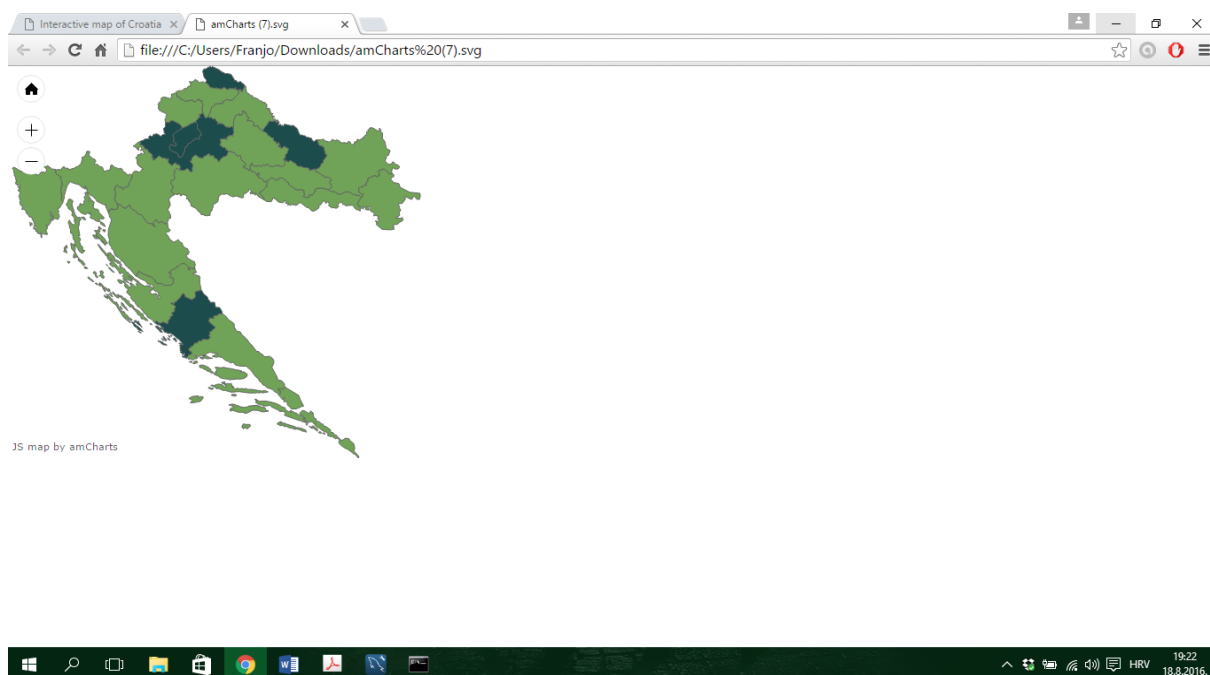
Izvor: autor

Karta se može spremiti u png, jpg, svg i pdf formatu. Na sl. 47. i sl. 48. karta se sprema u SVG formatu.



Slika 47. Spremanje karte u SVG obliku 01

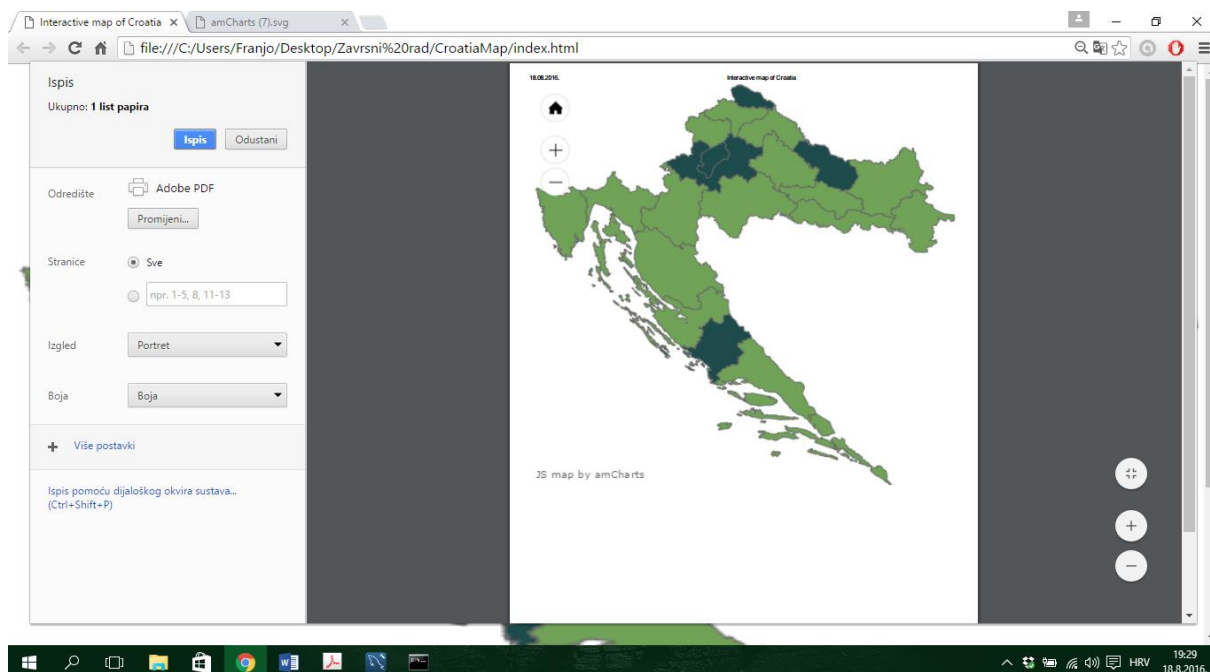
Izvor: autor



Slika 48. Spremanje karte u SVG obliku 02

Izvor: autor

Odabirom opcije "print" pojavljuje se prozor za ispis karte s dodatnim opcijama za ispis.



Slika 49. Ispis karte

Izvor: autor

14. Zaključak

SVG je W3C standard koji ne zahtjeva skupe alate za kreiranje jer ima standard otvorenog tipa. Baziran je na XML-u, a njime se može manipulirati uz pomoć JavaScript-a i drugih tehnologija. SVG je kompatibilan sa GIF-om, JPEG-om, PNG-om, HTML-om, CSS-om, SMIL-om i mnogim drugim standardima. Kreirane grafike u SVG-u su skalabilne i ne gube kvalitetu kod uređaja, stoga ih se može koristiti i na mobilnim uređajima. Grafika u SVG-u se jednostavno može kreirati pomoću poznatih vektorskih grafičkih programa ili pomoću tekst editora.

Iz istraživanja i izrade primjera, može se zaključiti da je SVG efektivna komponenta za izradu, animaciju i interaktivnost web grafike te jedan od najboljih izbora za to područje. SVG bolje radi s manjim brojem objekata, a jedna od glavnih značajki je mogućnost interaktivnosti svih elemenata. Moguće je umetanje i prikazivanje rasterskog sadržaja unutar SVG-a. Zbog vektorskog zapisa grafike, datoteke su iznimno male veličine. SVG zapis je tekstualan i otvoren pa se lako mogu mijenjati ili dodavati elementi i njegovi atributi.

U ovome završnom radu izrađena je funkcionalna aplikacija karta Hrvatske. Karta je interaktivna, što omogućuje povećavanje i smanjivanje karte, selektiranje županija, promjenu boja županija i promjenu broja stanovnika županije. Aplikacija ima opciju prikaza karte u pdf, png, jpg i SVG formatu. Svi moderni pretraživači dobro podržavaju ovu aplikaciju. Sučelje aplikacije je pojednostavljeno i nije teško za korištenje. Ovakav tip aplikacije jest vrlo koristan i može služiti u edukativne svrhe ili kao smjernica posjećenih dijelova Hrvatske s mogućnošću dodavanja još mnogih opcija.

Pomoću SVG-a može se kreirati grafika koja prikazuje kristalno jasnu sliku, neovisnu o rezoluciji ekrana. Iako primjena SVG-a nije pogodna za svaku situaciju, najbolji je izbor za vektorske bazirane ilustracije na web-u.

Literatura

- [1] Programska podrška za dvodimenzijску grafiku na web-u.
<http://www.zemris.fer.hr/predmeti/rg/diplomski/08Rebrovic/vektorskiXmlZapis.html>
(8.7.2016)
- [2] Ciganović-Janković, D. (2013.) Interaktivni grafički objekti u standardu HTML5.
Završni rad br. 3197. Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
https://bib.irb.hr/datoteka/636549.Zavrni_rad.pdf (17.6.2016.)
- [3] Inkscape. <http://en.wikipedia.org/wiki/Inkscape> (20.5.2016)
- [4] Dailey, D.; Frost, J.; Strazzullo, D. (2012.) Building Web Applications with SVG. USA, Microsoft Press.
- [5] Top SVG Javascript Libraries Worth Looking At.
<http://www.farinspace.com/top-svg-javascript-libraries-worth-looking-at/2> (12.7.2016.)
- [6] Raphael-JavaScript Library. <https://dmitrybaranovskiy.github.io/raphael/> (13.7.2016.)
- [7] Scalable Vector Graphics for Web Browsers using Flash.
<https://code.google.com/p/svgweb/> (23.6.2016.)
- [8] Dive Into Dojo GFX. <https://www.sitepen.com/blog/2010/12/30/dive-into-dojo-gfx/>
(12.7.2016)
- [9] Best JavaScript Libraries for Creating Interactive Map.
<https://iprodev.com/best-javascript-libraries-for-creating-interactive-maps/> (14.7.2016.)
- [10] Eisenberg, J. D. (2002.) SVG Essentials. USA, O'Reilly Media.